SetThreadToken

Do not continue execution of a client request if the function fails.

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-16

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5308 bytes

Attack Category	• Impersonation
Vulnerability Category	• Unchecked Return Value
Software Context	• Threads and Processes
Location	• winbase.h
Description	The SetThreadToken function assigns an impersonation token to a thread. The function can also cause a thread to stop using an impersonation token. Do not continue execution of a client request if the function fails.If the call to this function fails for any reason, the client is not impersonated and the client request is made in the security context of the calling process. If the calling process is running as a highly privileged account, such as LocalSystem, or as a member of an administrative group, the user may be able to perform actions that would otherwise be disallowed.
APIs	**Function Name** / **Comments** SetThreadToken
Method of Attack	An attacker could run a program that uses this function without checking the return value. When the check of the attacker's privileges fails, the program would continue to execute using its own privileges, rather than the attacker's (presumably) less privileges. This could result in the disclosure or alteration of sensitive information.
Exception Criteria	When the return value is checked and properly handled, this function is okay to use.
Solutions	**Solution Applicability** / **Solution Description** / **Solution Efficacy** Generally applicable to all invocations of SetThreadToken. / Always check the results of a SetThreadToken() / Effective.

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	call before using the token.
Signature Details	BOOL SetThreadToken(PHANDLE Thread,HANDLE Token);

Examples of Incorrect Code	`/* No error checking:`
	`* thread and token are both previously defined`
	`*/`
	`SetThreadToken(thread, token);`
	`//Continue privileged operation`
	
	`/* Proper error checking`
	`* thread and token are both previously defined`
	`*/`
	`if (!SetThreadToken(thread, token)) //Check to see whether this fails`
	`return false;`
	`//Continue privileged operation`

Examples of Corrected Code	
Source References	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/appsec.asp[2] • Howard, Michael & LeBlanc, David C. *Writing Secure Code, 1st ed*. Redmond, WA: Microsoft Press, 2002, ISBN: 0735615888. Chapter 16, “General Good Practices,” pg. 420. • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/setthreadtoken.asp[3] • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iissdk/html/2e68b0b5-341a-40e5-ad7c-1fa84b8b3b07.asp[4] • Spinellis, Diomidis. Software Security[5] (2005).
Recommended Resource	
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com